

UNITED STATES PATENT APPLICATION

FOR

**SYSTEM AND METHOD FOR SYNCHRONOUS CONFIGURATION OF DYNAMIC  
HOST CONFIGURATION PROTOCOL (DHCP) SERVER AND ROUTER  
INTERFACES**

**INVENTORS:**

**JEREMY ROVER  
AMBER SISTLA**

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP  
12400 WILSHIRE BOULEVARD  
SEVENTH FLOOR  
LOS ANGELES, CA 90025-1026

(503) 684-6200

EXPRESS MAIL NO.

EV325526569US

# SYSTEM AND METHOD FOR SYNCHRONOUS CONFIGURATION OF DHCP SERVER AND ROUTER INTERFACES

## RELATED APPLICATIONS

**[0001]** This nonprovisional patent application is related to the following contemporaneously filed nonprovisional patent applications: U.S. Patent Application serial number <042390.P17059>, entitled, “System and Method for Dynamically Configuring and Transitioning Wired and Wireless Networks;” U.S. Patent Application serial number <042390.P17061>, entitled “System and Method for Describing Network Resource Availability and Associations;” U.S. Patent Application serial number <042390.P17060>, entitled “System and Method for Programmatically Changing the Network Location of a Network Component”; and U.S. Patent Application serial number <042390.P17063>, entitled “System and Method for the Design and Description of Networks.”

## TECHNICAL FIELD

**[0002]** Embodiments of the invention generally relate to the field of networks and, more particularly, to a system and method for synchronous configuration of DHCP server and router interfaces.

## BACKGROUND

**[0003]** Mobile networking technologies are driving an evolution in the use and structure of networks. For example, users of mobile networking technologies expect to stay connected as they move from place to place and from network to network. Furthermore, users of mobile networking technologies expect easy and seamless network interface transitions as they move from place to place.

**[0004]** The term network component broadly refers to a node (e.g., a desktop, laptop, etc.) or a collection of nodes (e.g., a virtual private network, a subnet, a virtual local area network, etc.). The term node refers to a network component having a network interface. Examples of a node include switches, routers, servers, clients, workstations, laptops, handhelds, printers, hubs, and the like.

**[0005]** The movement of network components from place to place and network to network fosters constantly changing network infrastructures and topologies. Network components are typically configured to interact with particular network infrastructures and topologies. The term configuration can be used with respect to a network component or to an entire network. When used in association with a network component, configuration refers to the settings of software, hardware, and firmware that enable the network component to exchange information with a network. In a broader sense, configuring a network refers to configuring a plurality of network components to exchange information with one other.

**[0006]** Modern networking technologies increase the variety of network components that interact with a network and, also, the frequency at which these interactions occur. These interactions produce a combinational explosion of heterogeneous networks composed of many different network components each having a distinct configuration. This combinational explosion of heterogeneous networks is further complicated by the possibility that an initial network configuration changes over time as network components are added and removed from the network.

**[0007]** The term subnetwork (subnet) refers to a network that is a part of a larger network. Dividing networks into subnets is a well-known strategy to, for example,

increase the performance and enhance the security of a network. Typically, network components within a subnet exchange packets with network components in other subnets (or other networks) through a router.

**[0008]** FIG. 1 is a block diagram of a router 115 interconnecting network 100 with the Internet 120. Network 100 includes subnet 105 and subnet 110. Subnet 105 includes nodes 125 and 130, as well as Dynamic Host Configuration Protocol (DHCP) server 160. Similarly, subnet 110 includes nodes 135 and 140. A DHCP server refers to a network component that provides network administrative services in compliance with Request For Comments 2131 entitled, "Dynamic Host Configuration Protocol," R. Droms, March 1997. Router 115 connects subnet 105 and subnet 110 to Internet 120 and to each other. Router 115 includes router interfaces 145, 150, and 155. Each router interface is configured to provide the appropriate interface. For example, router interfaces 150 and 155 are configured with the IP addresses for subnets 105 and 110, respectively. Typically, a network administrator manually configures router interfaces 145, 150, and 155 in a time consuming and error prone process.

**[0009]** DHCP server 160 provides network administrative services to nodes 125 and 130 through DHCP server interface 165. For example, DHCP server 160 informs nodes 125 and 130 that router interface 150 is the gateway through which nodes 125 and 130 reach Internet 120 and subnet 110. In conventional network 100, DHCP interface 165 is manually configured in a slow and error prone process that is analogous to configuring router interface 150.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[00010]** Embodiments of the invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

**Figure 1** is a block diagram of a router 115 interconnecting network 100 with Internet 120.

**Figure 2** is block diagram of network 200 implemented according to an embodiment of the invention.

**Figure 3** is block diagram illustrating an embodiment of the invention abstracted into four layers.

**Figure 4** illustrates selected elements of exemplary network resource and association file 400.

**Figure 5** is an illustration of an exemplary current network state snapshot 500.

**Figure 6** is an exemplary illustration of network resource wrapper function call 600 with possible function parameters.

**Figure 7** illustrates exemplary network configuration request 700.

**Figure 8** is a conceptual illustration of selected interactions between abstract functional layers in network 800, according to an embodiment of the invention.

**Figure 9** is a flow diagram illustrating certain aspects of a method for configuring a router interface and a corresponding Dynamic Host Configuration Protocol (DHCP) server interface, according to an embodiment of the invention.

**Figure 10** is a simplified block diagram of selected elements of exemplary node 1000, implemented according to an embodiment of the invention.

**Figure 11** is a block diagram of selected elements of exemplary network 1100, implemented according to an embodiment of the invention.

## DETAILED DESCRIPTION

[00011] Embodiments of the invention enable quick and accurate dynamic configuration of corresponding router and Dynamic Host Configuration Protocol (DHCP) server interfaces. The ability to programmatically configure corresponding router and DHCP server interfaces is especially important in networks that are constantly changing (e.g., networks that contain mobile nodes). Embodiments of the invention ensure that DHCP server interfaces and router interfaces associated with the same subnet are configured with, for example, the appropriate IP address and gateway information.

[00012] FIG. 2 is block diagram of network 200 implemented according to an embodiment of the invention. FIG. 2 includes network component 210, router 220, DHCP server 230, subnet 240, and network information 250. A person of ordinary skill in the art will appreciate network 200 may include more and/or different components than those shown in FIG. 2. It is not necessary, however, that all of these generally conventional components be shown in order to disclose an illustrative embodiment for practicing the invention.

[00013] Network component 210 broadly represents any network component with sufficient processing resources to programmatically configure router 220 and DHCP server 230. In an embodiment of the invention, network component 210 is physically located proximate to router 220 and/or DHCP server 230. In an alternative embodiment of the invention, network component 210 is located at a remote site and connects with router 220 and DHCP server 230 through, for example, a Telnet session. In yet other alternative embodiments of the invention, network component 210 is DHCP server 230.

An exemplary embodiment of network component 210 is further described below with respect to FIG. 10.

**[00014]** Router 220 provides packet-forwarding services to network 200, in an embodiment of the invention. In an embodiment of the invention, router 220 executes packet forwarding based on layer 3 criteria. In alternative embodiments of the invention, router 220 executes packet forwarding based on both layer 3 and layer 2 criteria. Router interface 225 may be configured to provide packet-forwarding functions to subnet 240. The configuration of router interface 225 is stored in an electronic file, in an embodiment of the invention (e.g., router section 415, shown in FIG. 4). Routers are well known in the art and will not be further described except as to how they relate to embodiments of the invention.

**[00015]** DHCP server 230 provides network administrative functions in an embodiment of the invention. For example, DHCP server interface 235 may provide IP addresses, subnet masks, and/or gateway information to network components of subnet 240. The description for DHCP server interface 235 may be stored in an electronic file (e.g., DHCP server section 420, shown in FIG. 4). DHCP servers are well known to those of ordinary skill in the art and will not be further described except as to how they relate to embodiments of the invention. In alternative embodiments of the invention, network administrative services may be provided by a network component that complies with Request For Comments 951, entitled, "Bootstrap Protocol (BootP)," B. Croft, J. Gilmore, September 1985.



**[00016]** Network information 250 (may be derived in part from the network resource and association file and/or a virtual map) describes resources within network 200 and, also, the relationships between those resources. The illustrated embodiment of network information 250 includes: subnet IP address section 260, router section 270, and DHCP server section 280. Related U.S. Patent Application serial number <042390.P17061> further describes network resource and association files.

**[00017]** Network component 210 references information (e.g., a virtual map, which may be derived in part from the network resource and association file), to discover whether there is a router and/or a DHCP server associated with network 200, in an embodiment of the invention. Network information 250 is stored in memory and accessed by a processor (not shown) on network component 210, in an embodiment of the invention. In alternative embodiments of the invention, network information 250 is stored on a separate network component of network 200. In yet other alternative embodiments of the invention, network information 250 may be distributed over a plurality of network components or may be provided to network component 210 from a source outside of network 200.

**[00018]** Network component 210 may discover, for example, router 220 and DHCP server 230. Network component 210 may then determine whether router 220 and/or server 230 have an interface associated with subnet 240. In the illustrated embodiment of the invention, network component 210 discovers that router interface 225 and DHCP interface 235 are associated with subnet 240.

[00019] After discovering router interface 225 and DHCP interface 235, network component 210 determines an Internet Protocol (IP) address to assign to each interface. Determining an IP address to assign to each interface depends on the IP address naming convention used within a particular network. In an embodiment of the invention, the IP address scheme includes dividing IP address into prefixes and suffixes. The suffix for each IP address may be stored with the corresponding interface in network information 250. The prefix of each IP address may be the network IP address, in an embodiment of the invention. In an embodiment of the invention, network component 210 determines the IP address for each interface by determining the network IP address (e.g., by using a network mask) and combining the network IP address with the corresponding suffix for the respective interfaces stored in network information 250. A person of ordinary skill in the art appreciates that alternative IP addressing schemes may be used and alternative methods of deriving the appropriate IP address may be used in alternative embodiments of the invention.

[00020] In an embodiment of the invention, network component 210 determines additional configuration information. For example, network component 210 determines a gateway IP address for subnet 240, in an embodiment of the invention. In the illustrated embodiment of the invention, the gateway IP address is the same as the IP address for router interface 225. In alternative embodiments of the invention, a different and/or additional gateway addresses may be determined for subnet 240. After determining the appropriate configuration information, network component 210 programmatically configures router 220 and DHCP server 230 with the determined configuration

information. Programmatically configuring router 220 and DHCP 230 is further described below with regards to FIG. 9.

#### OVERVIEW OF THE FUNCTIONAL LAYERS

**[00021]** To provide the above stated functions, embodiments of the invention may be abstracted into four layers: the control layer, the network management layer, the verification and validation layer, and the physical network layer. FIG. 3 is block diagram illustrating an embodiment of the invention abstracted into four layers. In alternative embodiments of the invention, the functions may be abstracted into more layers or fewer layers. FIG. 3 includes control layer 310, network management layer 320, verification and validation layer 330, and physical network layer 340.

**[00022]** Control layer 310 may provide a single control point for functions provided by embodiments of the invention. Control layer 310 may be accessed through a console directly on a node in close proximity to the network or through a remote login session (e.g., Telnet). The functions of control layer 310 include generating network scenarios and directing the other layers to configure and transition the network based on the generated network scenarios, in an embodiment of the invention. Network scenarios may be generated randomly or may be based on predefined network configurations, in an embodiment of the invention. Also, control layer 310 may generate a series of network scenarios back-to-back. Each series of network scenarios can be reproduced by supplying a seed logged in past scenarios, in an embodiment of the invention.

**[00023]** Control layer 310 determines the current physical layout and state of the network based on its interactions with network management layer 320 and verification

and validation layer 330, in an embodiment of the invention. As will be further described below, control layer 310 accesses network management layer 320 to perform network configurations and network transitions. Transitioning a node broadly refers to, for example, transitioning a node from a first network interface to a second interface, and/or from a first subnet to a second subnet, and/or from a first Virtual Local Area Network (VLAN) to a second VLAN, and/or from a first topology to a second topology. The term network transition refers to transitioning one or more nodes within the network.

**[00024]** Network management layer 320 provides a number of functions including network configurations, network transitions, and maintenance of current network state information, in an embodiment of the invention. Network management layer 320 may be a stand-alone component for managing and reconfiguring network components. In alternative embodiments of the invention, network management layer 320 functions in association with the other layers illustrated in FIG. 3. In such an embodiment, the other layers may obtain network state information from network management layer 320.

**[00025]** FIG. 4 illustrates selected elements of exemplary network resource and association file 400. Network resource and association file 400 includes: dynamic network device section 402, non-dynamic network device section 404, power management device section 406, hubs section 408, Virtual Local Area Network (VLAN) switch section 410, router section 412, Dynamic Host Configuration Protocol (DHCP) server section 414, and addressing scheme section 416. A DHCP server refers to a network component that provides network administrative services in compliance with Request For Comments 2131 entitled, "Dynamic Host Configuration Protocol," R. Droms, March 1997. As illustrated in FIG. 4, network resource and association file 400

describes available network resources and associations in a standardized syntax. Related U.S. Patent Application serial number <042390.P17061> further describes network resource and association files.

**[00026]** Network management layer 320, in an embodiment of the invention, is responsible for configuring networks and transitioning networks. As is further discussed below in regard to FIG. 6, network resource wrappers 322, 324, and 326 may be used to configure and transition networks. In addition, network management layer 320 may maintain IP address allocation for network components and generate a readable text file that reports each IP address to facilitate communication across the network. After each network configuration and/or transition, network management layer 320 may generate a snapshot of the current network state.

**[00027]** FIG. 5 is an illustration of an exemplary current network state snapshot 500. The illustrated embodiment of current network state snapshot 500 is organized according to one or more subnets sections (e.g., subnet section 505). Each subnet section contains information about one or more network components within the subnet (e.g., node section 510). Node section 510 includes information about potential movement. Information about potential movement may include a list of network topologies based on network topologies and interfaces available to the node. For example, if a node contains an 802.11a network adaptor (and an 802.11a access point exists on the network), the node is able to make a transition to a wireless network connection. Network state snapshot 500 is more fully described in related U.S. Patent Application serial number <042390.P17063>. An 802.11a network adaptor and an 802.11a access point respectively refer to a network adaptor and an access point that comply with the IEEE

802.11 standard, entitled “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999 Edition.”

[00028] After a network transition, multiple sections of network state snapshot 500 may be updated since some network transitions affect multiple network components. Network management layer 320 uses network resource wrappers to programmatically configure network components, in an embodiment of the invention. Network resource wrappers abstract the functionality of a network component in a standardized way that allows network components to be interchanged when the components provide the same (or similar) functionality, in an embodiment of the invention. The term programmatically broadly refers to action performed by a software, hardware, and/or firmware resource of one or more network components.

[00029] FIG. 6 is an exemplary illustration of network resource wrapper function call 600. Network resource wrapper function call 600 may be used, for example, to configure a router. A person of ordinary skill in the art appreciates that similar network components may be configured with similar network resource wrappers. In an embodiment of the invention, there is a network resource wrapper corresponding to each configurable network component in a network. Table 1 provides a description of the fields of exemplary network resource wrapper function call 600.

TABLE 1

Field	Description
IpAddr 605	IpAddr 605 is the IP address from which the router can be configured, in an embodiment of the invention.
Passwd 610	Passwd 610 may be used to, for example, enable a Telnet session to the router.
IntfType 615	IntfType 615 represents the type of interface to be modified (e.g., Ethernet) on the router.
IntfNum 620	IntfNum 620 is the number of the interface to be modified, in

	an embodiment of the invention.
IntfIP 625	IntfIP 625 provides the new IP address for the interface, in an embodiment of the invention.
SubnetMask 630	SubnetMask 630 provides the subnet mask for the subnet with which the interface is associated.
Ext 635	Ext 635 indicates whether the interface is “inside” or “outside” of a VPN, in an embodiment of the invention.
PermitIP 640	PermitIP 640 represents IP addresses that are permitted on the interface, in an embodiment of the invention.

**[00030]** Referring again to FIG. 3, verification and validation layer 330 abstracts all devices that are used to verify and validate the current network configuration. These devices may include, for example, packet sniffers, traffic generators, and other network validation devices. In an embodiment of the invention, third party verification tools and/or proprietary tools can be added to this layer to provide seamless accessibility to a wide range of network analysis and traffic generation tools.

**[00031]** In an embodiment of the invention, the network validation devices are mobile. For example, during network scenario execution, network validation devices may be placed on the particular subnet that requires validation. In contrast, conventional network validation typically involves the manual movement of validation devices from one subnet to another, and/or one VLAN to another, and/or one network interface to another.

**[00032]** Control layer 310 interoperates with verification and validation layer 330 to perform graceful state recovery, in an embodiment of the invention. During the execution of a network scenario, verification and validation layer 330 detects and logs results to report to the control layer. Control layer 310 determines whether to perform graceful state recovery based on the errors, if any, logged and reported by the verification

and validation layer. Graceful state recovery refers to reconfiguring network components to a state they were in before an error occurs during the execution of a network scenario.

**[00033]** Physical network layer 340 contains the physical network devices of network 300 (e.g., physical network resources 342, 344, and 346). In an embodiment of the invention, physical network resources 342, 344, and 346 correspond to network resource wrappers 322, 324, and 326 described above with respect to FIG. 6. Control layer 310 may request the functionality of physical network resources 342, 344, and 346 through a network configuration request.

**[00034]** FIG. 7 illustrates exemplary network configuration request 700. Network configuration request 700 includes subnet group section 705 and device section 710. Subnet group section 705 may be used to organize a plurality of subnet subsections (e.g., subnet subsection 715). Each subnet subsection may list information about the type of network topology requested for the subnet. For example, a particular subnet may include both wired and wireless network topology. Device section 710 may include information about nodes within a requested network configuration and a start position for mobile nodes within a requested network configuration. Network configuration requests are more fully described in related U.S. Patent Application serial number <042390.P17063>.

**[00035]** In operation, control layer 310 may send a network configuration request to network management layer 320. Network management layer 320, in turn, may use network resource wrappers to programmatically configure the physical resources in physical network layer 340. When new physical resources are added to network 300, corresponding network wrappers may be written to network management layer 320 to abstract the functionality of the new resource. In an embodiment of the invention, non-



configurable network resources may be added and/or removed from network 300 at will because network resource wrappers are not needed to interact with non-configurable network resources.

#### INTERACTIONS AMONG THE LAYERS

**[00036]** FIG. 8 is a conceptual illustration of selected interactions between abstract functional layers in network 800, according to an embodiment of the invention. Network 800 includes control layer 802, network management layer 804, physical network layer 806, and verification and validation layer 808. A person of ordinary skill in the art will appreciate that, in alternative embodiment embodiments of the invention, network 800 may include more layers or fewer layers.

**[00037]** A user may provide an input to initiate a network scenario, in an embodiment of the invention (not shown). Control layer 802 queries network management layer 804 to determine if executing the network scenario is possible given the current network configuration at 810. If the network scenario is supported in the current network configuration, control layer 802 initiates network verification and validation at 812.

**[00038]** Otherwise control layer 802 resolves the network scenario into a network configuration and creates a corresponding network configuration request at 814. The network configuration request may contain one or more subnets as well as the starting position on the network for mobile nodes. Network management layer 804 configures the network at 816 and reports success or failure of the configuration at 818. If network management layer 804 does not report any failures occurring during the configuration

process, control layer 802 triggers the verification and validation layer 808 at 812.

Verification and validation layer 808 performs network verification and/or validation tests and reports the findings to control layer 802 at 820.

**[00039]** A network scenario may include transitioning one or more nodes. For example, a network scenario may include transitioning a node from a wired LAN connection to a wireless LAN connection. Control layer 802 queries network management layer 804 to determine whether a transition is supported by the network configuration at 822. If the transition is supported then control layer 802 requests the transition at 824. Network management layer 804 reports success or failure of the transition at 826. If the transition is successful, control layer 802 prompts verification and validation layer 808 to perform appropriate tests at 828 and report the findings to control layer 802 at 830.

**[00040]** Embodiments of the invention may iterate the network configuration and transition processes to enable multiple network scenarios to occur one after another. Alternatively, the network scenario process may terminate after a single iteration. In such an embodiment, the findings of the completed network scenario may be reported to a user and a pseudo-random seed to reproduce the network scenario may be stored.

**[00041]** Turning now to FIG. 9, the particular methods associated with embodiments of the invention are described in terms of computer software and hardware with reference to a flowchart. The methods to be performed by a control layer and/or a management layer may constitute state machines or computer programs made up of computer-executable instructions. Describing the methods by reference to a flowchart enables one of ordinary skill in the art to develop such programs including such

instructions to carry out the methods on suitably configured computing devices (e.g., one or more processors of a network component) executing the instructions from computer-accessible media. The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, embodiments of the invention are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, process, procedure, agent, application, etc.), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computing device causes the device to perform an action or produce a result. For ease of discussion, the entities performing the functions of each layer are hereinafter referred to as agents. For example, the entity (or entities) performing the functions of the management layer is referred to as the management agent. An agent may be executable content, control logic, firmware, or some combination thereof, in an embodiment of the invention.

**[00042]** FIG. 9 is a flow diagram illustrating certain aspects of a method for configuring a router interface and a corresponding Dynamic Host Configuration Protocol (DHCP) server interface. Referring to process block 910, a network management agent (not shown) references information (e.g., a virtual map, which may be derived, in part, from the network resource and association file) to determine whether a DHCP server

interface and a router interface are associated with the same subnet. The network management agent may reference a file of network components that may include addressing scheme information to identify the addressing scheme used by the listed network components. The referenced file has a standardized syntax in an embodiment of the invention. Network information 250, shown in FIG. 2 and network resource and association file 400 shown in FIG. 4 are examples of electronic files that list network components with a standardized syntax. Network resource and association files are more fully described in related U.S. Patent Application serial number <042390.P17061>.

[00043] In an embodiment of the invention, the network management agent resides on a (DHCP) server. A DHCP server refers to a server that is compliant with Request for Comments 1541, "Dynamic Host Configuration Protocol," R. Droms, October 1993. In an alternative embodiment of the invention, the network management agent may reside on a control node. A person of ordinary skill in the art appreciates that the network management agent may reside on any of a number of different network components or may be distributed among a number of network components.

[00044] Referring to process block 920, in an embodiment of the invention, the network management agent determines configuration information for the DHCP server interface based, at least in part, on information in the referenced electronic file. Determining configuration information broadly refers to determining information to enable the DHCP server interface to provide network administrative services to network components within a subnet. The network management agent may determine an IP address for the DHCP server interface based on the addressing scheme information in the referenced file. The network management agent may also determine a gateway IP

address to distribute to DHCP clients within the subnet. In the illustrated embodiment of the invention, the gateway IP address is the IP address of the corresponding router interface.

**[00045]** Referring to process block 930, in an embodiment of the invention, the network management agent determines configuration information for the router interface based, at least in part, on information in the referenced electronic file. Determining configuration information broadly refers to determining information to enable the router interface to provide a gateway from the subnet to other subnets and/or the Internet. Determining configuration information may include, for example, determining an IP address for the router interface so that the router interface can serve as a gateway for the subnet.

**[00046]** Referring to process block 940, in an embodiment of the invention, the network management agent programmatically configures the DHCP server with the determined configuration information. For example, the network management agent may call a function (e.g., network resource wrapper function call, shown in FIG. 6) to interact with the DHCP server and change its state. The network management agent may pass configuration information to the called function, which may then programmatically configure the DHCP server. For example, the network management agent may pass the IP address of the router interface, the subnet mask, the gateway IP address, and/or other information to the called function.

**[00047]** Referring to process block 950, in an embodiment of the invention, the network management agent programmatically configures the router with the determined configuration information. For example, the network management agent may call a

function (e.g., network resource wrapper function call, shown in FIG. 6) to interact with the router and change its state. The network management agent may pass configuration information to the called function, which may then programmatically configure the router. For example, the network management agent may pass the Internet Protocol address of the DHCP server interface, the subnet mask, and/or other information to the called function.

**[00048]** FIG. 10 is a simplified block diagram of selected elements of exemplary node 1000, implemented according to an embodiment of the invention. Node 1000 may include: one or more processor(s) 1010, memory 1020, one or more Input/Output interfaces 1030, network interface(s) 1040, control agent 1050, management agent 1060. The illustrated elements may be connected together through system interconnect 1070. Processor(s) 1010 may include a microprocessor, microcontroller, field programmable gate array (FPGA), application specific integrated circuit (ASIC), central processing unit (CPU), programmable logic device (PLD), and similar devices that access instructions from system storage (e.g., memory 1020), decode them, and execute those instructions by performing arithmetic and logical operations. In some embodiments of the invention, processor(s) 1020 is implemented with a plurality of processors.

**[00049]** Memory 1020 may encompass a wide variety of memory devices including read-only memory (ROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), random access memory (RAM), non-volatile random access memory (NVRAM), cache memory, flash memory, and other memory devices. Memory 1020 may also include one or more hard disks, floppy disks, ZIP disks, compact disks (e.g., CD-ROM), digital

versatile/video disks (DVD), magnetic random access memory (MRAM) devices, and other system-readable media that store instructions and/or data. Memory 1020 may store program modules such as routines, programs, objects, images, data structures, program data, and other program modules that perform particular tasks or implement particular abstract data types that facilitate system use.

**[00050]** One or more I/O interfaces 1030 may include a hard disk drive interface, a magnetic disk drive interface, an optical drive interface, a parallel port, serial controller or super I/O controller, serial port, universal serial bus (USB) port, a display device interface (e.g., video adapter), a sound card, modem, and the like.

**[00051]** Network interface(s) 1040 may include a wide variety of software, hardware, and/or firmware to interface node 1000 with an associated network (not shown). In an embodiment of the invention, network interface 1040 includes both wired (e.g., local area network) interfaces and wireless (e.g., wireless local area network) interfaces. Network interface(s) 1040 may include network interface card(s) and/or chipsets that provide a network interface.

**[00052]** Control agent 1050 enables node 1000 to act as a single control point for a network to which node 1000 is connected. Control agent 1050 may be executable content, control logic (e.g., ASIC, PLD, FPGA, etc.), firmware, or some combination thereof, in an embodiment of the invention. In embodiments of the invention in which control agent 1050 is executable content, it may be stored in memory 1020 and executed by processor(s) 1010.

**[00053]** Management agent 1060 enables node 1000 to perform network configuration changes and network transitions, in an embodiment of the invention.

Management agent 1060 may be executable content, control logic (e.g., ASIC, PLD, FPGA, etc.), firmware, or some combination thereof, in an embodiment of the invention. In embodiments of the invention in which management agent 1060 is executable content, it may be stored in memory 1020 and executed by processor(s) 1010. In the illustrated embodiment of the invention, management agent 1060 resides on the same node as control agent 1050. In alternative embodiments of the invention, control agent 1050 and management agent 1060 reside on separate nodes. In yet other alternative embodiments of the invention, control agent 1050 and/or management agent 1060 are distributed across more than one node.

**[00054]** System interconnect 1070 permits communication between the various elements of node 1070. System interconnect 1070 may include a wide variety of signal lines including one or more of a memory bus, peripheral bus, local bus, host bus, bridge, optical, electrical, acoustical, and other propagated signal lines.

**[00055]** FIG. 11 is a block diagram of selected elements of exemplary network 1100, implemented according to an embodiment of the invention. Network 1100 includes control node 1105, DHCP server 1110, router 1115, VLAN switch 1120, Virtual Private Network (VPN) 1125, hub 1130, and node 1135, power switch serial controller device 1140, and access point 1145.

**[00056]** Control node 1105 provides a single control point for executing network configurations, network transitions, and/or network scenarios, in an embodiment of the invention. A control agent (e.g., control agent 1050 shown in FIG. 10) resides on control node 1105 in an embodiment of the invention. In alternative embodiments of the invention, a control agent and a management agent (e.g., management agent 1040 shown



in FIG. 10) reside on control node 1105. Control node 1105 may be a general purpose computing device containing a control agent, in an embodiment of the invention.

**[00057]** DHCP server 1110 provides network administrative functions in an embodiment of the invention. For example, DHCP server 1110 may provide IP addresses, subnet masks, and/or gateway information to network components of network 1100. The DHCP server may associate one or more network interfaces with corresponding IP address information (e.g., IP address, subnet mask, and gateway). The associations between network interfaces and corresponding IP address information determine which nodes receive which network administrative functions, in an embodiment of the invention. DHCP servers are well known to those of ordinary skill in the art and will not be further described except as to how they relate to embodiments of the invention. In an embodiment of the invention a network management agent resides on DHCP server 1110. In such an embodiment of the invention, DHCP server 1110 may be referred to as a management node. The term management node broadly refers to a node on which a management agent (or a portion of a management agent) resides.

**[00058]** Router 1115 provides a number of network interfaces in an embodiment of the invention. Each network interface may be associated with IP address information (e.g., interface IP address and subnet) to enable the exchange of packets with the interface. Routers are well known to those of ordinary skill in the art and will not further described except as to how they relate to embodiments of the invention.

**[00059]** VLAN switch 1120 provides a plurality of ports and supports a plurality of VLANs, in an embodiment of the invention. Each supported VLAN may include one or more ports. Each port may be connected to one or more network components. VLAN

switch 1120 enables an embodiment of the invention to group hubs together programmatically into logical subnets. VLAN switches are well known to those of ordinary skill in the art and will not be further described except as to how they relate to embodiments of the invention.

[00060] VPN 1125 provides a mechanism for secure transactions in an embodiment of the invention. In some embodiments of the invention, one or more VPNs employ static IP address configurations. In such embodiments of the invention, a network management agent may create the specific subnets used to communicate with the statically configured VPN. This may be accomplished, for example, by configuring the IP addresses on the DHCP server with the subnet IP addresses that correspond to the specific VPN. In addition, router 1115 may be configured to isolate network traffic on either side of the VPN so that only VPN traffic is routed. Internal traffic refers to traffic within the VPN (or firewall) and external traffic refers to traffic outside of the VPN (or firewall). VPNs are well known to those of ordinary skill in the art and will not be further described except as to how they relate to embodiments of the invention.

[00061] It should be appreciated that reference throughout this specification to “one embodiment” or “an embodiment” means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Therefore, it is emphasized and should be appreciated that two or more references to “an embodiment” or “one embodiment” or “an alternative embodiment” in various portions of this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or

characteristics may be combined as suitable in one or more embodiments of the invention.

[00062] Similarly, it should be appreciated that in the foregoing description of exemplary embodiments of the invention, various features of the invention are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims following the detailed description are hereby expressly incorporated into this detailed description, with each claim standing on its own as a separate embodiment of this invention.

---